

1 REMARKS

2 Status of the Claims

3 Claims 1-56 are now pending in the present application; Claims 57-61 having been
4 previously canceled and Claims 1, 14, 21, and 35 having been amended to more clearly distinguish
5 the recited subject matter over the cited art. Claim 29 has been amended to correct a typographical
6 error.

7 Brief Summary of Telephone Interview

8 On May 30, 2007, applicants' attorney and Examiner Strange spoke regarding proposed
9 amendments to the claims that were faxed prior to the interview.

10 With respect to steps (c) and (d) of Claim 1, applicants' attorney explained why the machine
11 instructions of column 9, lines 5-13 that are appended to a Web page in U.S. Patent No 6,078,956
12 (Bryant) do not cause the client computer to actually determine the last_rsp_time variable. In support
13 of this position, applicants' attorney pointed out that column 2, line 66-column 3, line 2 of Bryant
14 states that the client software is provided in a computer-readable medium or downloaded over the
15 network. Based on this teaching in Bryant, the machine instructions that cause the client computer to
16 determine the response time are already present on the client side, prior to reception of the server web
17 page that includes the Java script. Examiner Strange disagreed and explained his belief that the Java
18 script appended to the Web page is the code that determines the performance metric.

19 Although no agreement was reached during the interview, applicants' attorney would like to again
20 thank Examiner Strange for his time and willingness to discuss these issues during the telephone interview.
21 After further review of the above-identified application and cited art following the telephone interview,
22 applicants are respectfully not persuaded that Examiner Strange is correct in his position, and have included
23 herein a written discussion of arguments presented during the telephone interview and a further reason
24 supporting applicants' position that the code for determining the response time are present on the client
25 computer prior to receipt of the Java code appended to a Web page, from the Web server. This existing
26 computer code on the client implements a test routine in connection with Bryant's FIGURE 4, as described
27 in detail below.

28 Claims Rejected Under 35 U.S.C. § 102

29 The Examiner has rejected Claims 1-4, 6, 7, 10-16, 18, 19, 21-24, 26, 29, 31-36, 38, 40-45,
30 47, 50, 52, and 53 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No 6,078,956

1 (Bryant et al., hereinafter referred to as “Bryant”). Applicants respectfully disagree for at least the
2 following reasons.

3 In the interest of reducing the complexity of the issues for the Examiner to consider in this
4 response, the following discussion focuses on independent Claims 1, 13, 14, 21, 35, and 42. The
5 patentability of each remaining dependent claim is not necessarily separately addressed in detail.
6 However, applicants’ decision not to discuss the differences between the cited art and each
7 dependent claim should not be considered as an admission that applicants concur with the
8 Examiner’s conclusion that these dependent claims are not patentable over the disclosure in the
9 cited references. Similarly, applicants’ decision not to discuss differences between the prior art and
10 every claim element, or every comment made by the Examiner, should not be considered as an
11 admission that applicants concur with the Examiner’s interpretation and assertions regarding those
12 claims. Indeed, applicants believe that all of the dependent claims patentably distinguish over the
13 references cited. Moreover, a specific traverse of the rejection of each dependent claim is not
14 required, since dependent claims are patentable for at least the same reasons as the independent
15 claims from which the dependent claims ultimately depend.

16 Patentability of Independent Claim 1

17 Significant differences exist between the cited art and Claim 1 because the cited art does not
18 teach or suggest that the one or more performance metrics for the distributed application are actually
19 determined by implementation of the performance monitoring function, as recited in step (d), *wherein*
20 *the performance monitoring function is defined by appended machine instructions*. A portion of step
21 (c) recites:

22 appending machine instructions that define a performance monitoring function to the
23 distributed application data

24 A portion of step (d) recites (emphasis added):

25 *executing the machine instructions* at the second site, to implement the performance
26 monitoring function and *to determine the one or more performance metrics* for the
27 distributed application

28 For example, with respect to FIGURE 3, the disclosure explains that:

29
30 To avoid the need for the user to explicitly install a performance monitoring program,
machine instructions that define a performance monitoring function are included with

1 the distributed application data that were requested and both the machine instructions
2 and the distributed application data are transmitted over the network to the first site.
3 The *machine instructions are then executed at the first site*, automatically and
4 transparently, to *implement the performance monitoring function* used to determine the
5 one or more performance metrics for the distributed application. (Specification,
6 page 8, lines 24-31 – emphasis added.)

7 Thus, it is clear from the italic font portions of this quote, that the performance monitoring
8 function is implemented by the machine instructions and that the performance monitoring function
9 determines the performance metric. However, as recited in step (c), these machine instructions are
10 the *appended* machine instructions that define the performance monitoring function, i.e., that indicate
11 how the actual determination of the performance metric is calculated.

12 In contrast, the function implemented by Bryant's appended machine instructions does not
13 actually indicate how to determine the performance metric, but only instructs as to *when* it should be
14 determined. The machine instructions in Bryant that actually determine the performance metric are
15 provided in a computer-readable medium or otherwise are separately downloaded to the client over
16 the computer network, as explained in further detail below. Thus, the machine instructions that
17 Bryant uses to actually determine a performance metric, such as the response time, are not the
18 appended instructions (i.e., are NOT submitted by the server to the client as part of the Web page
19 requested by the client). The machine instructions that Bryant indicates are appended to a Web page
20 are different, as reflected in column 9, lines 5-18 of Bryant.

21 In order to illustrate this point, it may be helpful to also review the Examiner's assertion of
22 equivalence to the performance metric as recited in applicants' step (d) and to the equivalence of
23 applicants' step (c) that recites:

24 appending machine instructions that define a performance monitoring function to the
25 distributed application data that were requested and transmitted over the network to
26 the second site as one data file; and

27 On page 3 of the Office Action, the Examiner asserts that the time required to service the
28 request (column 9, lines 5-18) is equivalent to the one or more performance metrics, and that the
29 equivalent of step (c) is disclosed in at least column 8, line 66 to column 9, line 15 because the code
30 is submitted as part of the page. This citation from Bryant is reproduced below:

If NETSCAPE.sub.-- LIVEWIRE.curr.sub.-- time on the client was set to 987666532,
then the rest of the statements in the <server> code above cause the following client-

side JavaScript to be submitted to the client as part of the page. (Bryant, column 8, line 66 – column 9, line 2.)

```
<SCRIPT>
```

```
if (document.cookie.indexOf("RSP.sub.-- TIME NOT.sub.-- SET") != -1) {  
document.cookie = "NETSCAPE.sub.-- LIVEWIRE.last.sub.-- rsp.sub.-- time+(new  
Date()).getTime() -987666532; document.cookie = "NETSCAPE.sub.--  
LIVEWIRE.rsp.sub.-- time.sub.-- set= RSP.sub.-- TIME.sub.-- SET;";  
document.cookie = "NETSCAPE.sub.-- LIVEWIRE.curr.time=-1;
```

```
</SCRIPT >
```

(Bryant, column 9, lines 5-14.)

The if statement ensures that last_rsp_time is updated only the first time that the "onload" action is invoked. The last_rsp_time contains the time in milliseconds required to service this request and deliver the page to the client. (Bryant, column 9, lines 15-19 – emphasis added.)

As indicated by the italic font portion above, Bryant discloses that the first statement ensures that a variable, *last_rsp_time*, is only updated the first time that the "onload" action is invoked. As further indicated by the italic font portion of the preceding quote, this variable contains the time required to service this request and deliver the page to the client.

However, notice that these machine instructions (i.e., instructions that ensure when the variable is updated, *last_rsp_time*) do NOT actually control how to determine the one or more performance metrics (i.e., time required to service this request) as recited in step (d). Instead, Bryant recites that:

Preferably, the present invention is implemented with a client "piece" and a server "piece." **The client software may be a computer program product implemented in a computer-readable medium or otherwise downloaded to the Web client over the computer network.** This software includes first program code means, responsive to a first HTTP request from the Web client to the Web server, for calculating the response time associated with the first HTTP request, and second program code means, responsive to the calculating means and a second HTTP request, for placing the response time in a cookie associated with the second HTTP request. The second HTTP request is then sent to the Web server to pass the response time information as previously described. (Emphasis added, Bryant, column 2, line 65-column 3, line 10.)

1 As indicated by the underlined portion above, and particularly, the bold/italic font portion, the
2 source of the machine instructions that actually determine the performance metric are either a
3 computer program product that is either on a computer-readable medium or which is separately
4 downloaded to the Web client over the computer network. However, it is clear that the source of the
5 machine instructions that actually specify how to determine the performance metric are NOT
6 submitted to the client as part of the Web page and thus are not included with the machine
7 instructions that are part of the Web page provided to the client, which are discussed in column 9,
8 lines 5-18 of Bryant. Indeed, in Bryant, the machine instructions that control how the client
9 determines the performance metric are already executing on the client before the Web page arrives
10 from the server. Notice that Bryant teaches the following approach for calculating the response time:

11 The response time associated with the first HTTP request is calculated without
12 synchronized client and server machine clocks or use of an external timebase. Rather,
13 the end user response time is calculated using just a clock in the Web client. *In*
14 *particular, this calculation begins by recording a first clock time on the clock, the first*
15 *time associated with the transmission of the first HTTP request from the Web client to*
16 *the Web server.* Upon receipt at the Web client of a specified event in a response to the
17 first HTTP request, a second clock time on the clock is recorded. The response time is
18 then generated by subtracting the first clock time from the second clock time. The
19 specified event in the response to the first HTTP request may be any event in the
20 response, such as first packet return, last HTML byte, last .gif byte, receipt of some
21 intermediate byte or page element, or the like (provided this is done consistently from
22 request to request). Because the "timebase" for the response time calculation is all
23 "local" (i.e. within the Web client itself), any given event may be used as the trigger
24 for the calculation. The calculated response time is saved and then sent to the Web
25 server upon a next HTTP request to that server. (Bryant, column 2, lines 44-64.)

26 It is clearly apparent that machine instructions to determine the performance metric must already be
27 executing on the client, before the first clock time associated with the transmission of the first HTTP
28 request from the Web client to the Web server can be recorded by the client. An event handling
29 routine on the client computer will not record a time when the first HTTP request was made unless
30 machine instructions are executed on the client computer that would cause it to do so. But if the
machine instructions (i.e., Java script) that causes the clock time to be recorded are indeed appended
to the Web page that is delivered in response to the first HTTP request (as asserted by the Examiner),
those machine instructions will not have been available on the client computer at the time they would
have to have been executed to cause the clock time of that first request to be recorded. As

1 highlighted in the italic font portion above, recordation of the first clock time takes place when the
2 first HTTP request is transmitted. Again, since these instructions would not have been available to
3 cause “the first time associated with the transmission of the first HTTP request from the Web client to
4 the Web server” to be recorded before the Web page is actually received, the Web page cannot
5 provide the machine instructions that cause that first clock time to be recorded. Instead, it is the
6 machine instructions that were included on a memory medium (e.g., on a CD) or which were
7 separately downloaded and installed on the client computer that control the recordation of the first
8 clock time and the subsequent determination of the performance metric.

9 The disclosure of the cited Bryant reference also teaches that “the present invention provides
10 a technique for calculating response time associated with a given HTTP request and then recording or
11 logging that response time at the Web server for subsequent analysis and use” (Bryant, column 5,
12 lines 21-26). A preferred embodiment is described in connection with FIGURE 4. Bryant also
13 discloses:

14 Thus, in accordance with the preferred embodiment, a response time associated with a
15 first HTTP request is calculated as the HTTP request is being processed but not passed
16 to the Web server until a subsequent HTTP request (typically, a next request).
17 Preferably, the response time information is passed to the Web server in a special
18 cookie associated with the subsequent HTTP request. A more detailed description of
19 this process is now provided. (Bryant, column 5, lines 47-54.)

20 As is apparent from this citation, although response time information is passed to the Web server, it is
21 not calculated by any instructions appended to a page that was requested by the client from the Web
22 server. In connection with the description of Bryant’s FIGURE 4, at a step 52, a test is run repeatedly
23 to determine whether a given HTTP request has been issued from the client to the server (Bryant,
24 column 5, lines 27-30). For any type of test to be run, it is apparent that machine instructions ***must***
25 ***pre-exist*** on the client side prior to reception of a Web page in order to determine the performance
26 metric for the delivery of that Web page.

27 For the reasons noted above, the rejection of Claim 1 as being anticipated by Bryant should be
28 withdrawn, since the claim is patentable over this reference. Similarly, since dependent claims
29 inherently include what is recited in the independent claim on which they depend, Claims 2-4, 6, 7,
30 and 10-12 are patentable over Bryant for at least the same reasons as set forth above in connection
with Claim 1.

1 Patentability of Independent Claim 13

2 Independent Claim 13 is directed toward a machine-readable medium on which are stored
3 machine instructions for appending to distributed application data that are transferred from one site to
4 another as one data file. Significant differences exist between the cited art and Claim 13 because the
5 cited art does not teach or suggest the one or more performance metrics for the distributed application
6 are determined by the performance monitoring function, as recited in subparagraph (b), wherein the
7 performance monitoring function is defined by the machine instructions *that are included with the*
8 *distributed application* transferred between sites. The Examiner has rejected Claim 13 under the
9 same rationale as Claim 1, and explains that they recite substantially identical subject matter.

10 A portion of subparagraph (b) recites (emphasis added):

11 *the performance monitoring function to determine one or more performance metrics*
12 *for a distributed application in which the distributed application data are transferred*
13 *between sites and accessed at one of the sites*

14 As highlighted in the italic font portion above, it is the performance monitoring function that
15 actually determines the one or more performance metrics, and the machine instructions for doing so
16 are included with the distributed application data file. Since applicants understand the Examiner to
17 be asserting that Bryant's time required to service a request (Office Action page 3) is equivalent to
18 applicants' performance metric, as explained above in connection with applicants' traversal of the
19 rejection of Claim 1, the function implemented by Bryant's machine instructions (which are included
20 with a Web page transferred to a client from a server) does not actually determine the performance
21 metric, but only instructs as to *when* it should be determined.

22 Thus, for reasons generally consistent with those noted above in applicants' traversal of the
23 rejection of independent Claim 1, independent Claim 13 also distinguishes over Bryant. For the
24 reasons noted above, the rejection of Claim 13 as being anticipated by Bryant should be withdrawn,
25 since the claim is patentable over these references.

26 Patentability of Independent Claim 14

27 Independent Claim 14 is directed toward a system for determining one or more performance
28 metrics for a distributed application in which distributed application data are transferred from a first
29 site to a second site over a network. Significant differences exist between the cited art and Claim 14
30 because the cited art does not teach or suggest that said at least one performance metric for the

distributed application are determined by the performance monitoring function that is defined by the appended machine instructions, as recited in subparagraph (b). A portion of subparagraph (b) has been amended to recite:

a performance monitoring function defined by the machine instructions such that when executed by said processing device as the distributed application data are accessed at the second site, *said performance monitoring function determining said at least one performance metric* and being implemented without requiring any affirmative action by a user

Applicants have clarified that the performance monitoring function is defined by the machine instructions as highlighted by the underlined portion above. In addition, as highlighted in the italic font portion above, the claim recites that the performance monitoring function determines at least one performance metric. On page 6 of the Office Action, the Examiner has asserted that Bryant's fetch latency is equivalent to applicants' at least one performance metric and that Bryant's code that is submitted as part of the page is equivalent to applicants' performance monitoring function. However, as explained above in connection with applicants' traversal of the rejection of Claim 1, the function implemented by Bryant's code that is submitted as part of the page does not actually determine the performance metric of the fetch latency, but only instructs *when* it should be determined.

For the reasons noted above, the rejection of Claim 14 as being anticipated by Bryant should be withdrawn, since the claim is patentable over these references. Similarly, since dependent claims inherently include what is recited in the independent claim on which they depend, Claims 15-16, 18, and 19, are patentable over Bryant for at least the same reasons as set forth above in connection with Claim 14.

Patentability of Independent Claim 21

Independent Claim 21 is directed toward a method for determining and collecting at least one performance metric related to access of a Web page by a browser program on a client device, including at least one of a compound performance metric and a correlated performance for a network. Significant differences exist between the cited art and Claim 21 because the cited art does not teach or suggest that the appended machine instructions that define a browser monitoring program as recited in step (c) are executed to carry out the browser monitoring function, such that said at least

1 one performance metric on the client device is determined with the browser monitoring function as
2 recited in step (d).

3 Below appear two citations from applicants' disclosure explaining that two types of
4 monitoring elements are a server monitor and a browser monitor:

5
6 *The measurement system of the present invention preferably includes two types of*
7 *monitoring elements.* The first type is a server monitor that resides on a server
8 responsible for the creation of content. Alternatively, the server monitor function
9 occurs in a Web caching proxy. In another alternative, both a server monitor and a
10 Web caching proxy monitor may be used. In general, the system benefits from
11 monitors on each of the devices of the network that impact the end-user experience,
12 with each monitor revealing a contribution of the device on which it is active, to the
13 overall performance experienced by the end user. *The second type is a browser*
14 *monitor that is implemented using a program delivered transparently to the end user*
15 *with a monitored Web page or other type of distributed application data.* However, it
16 may not be necessary for a Web browser program to load the browser monitor if there
17 is an up-to-date copy of the program in a cache accessible by the browser program. *In*
18 *other situations, the browser monitor is embedded directly in the managed Web page.*
19 Both performance monitors preferably deliver the performance data that they collect to
20 a central collection and reporting point or data center, for data analysis and generation
21 of related reports. Additional performance monitors can optionally be included to
22 provide management data from other intermediate perspectives in the system.
(Emphasis added, applicants' specification, page 12, line 15-page 13, line 2.)

19 The browser monitor checks for the presence of a server monitor cookie, in a step 104.
20 If the server monitor cookie is set, **the browser monitor** recognizes that this Web
21 page being loaded is a managed Web page and in response to this determination,
22 records various state information and **determines the performance metrics for the**
page in a step 106. (Emphasis added, applicants' specification, page 19, lines 21-25.)

23 As highlighted in the italic font portion of the first quote above, one of these monitoring
24 elements is a browser monitor, and as highlighted in the bold font portion of the second quote, the
25 browser monitor determines the performance metrics. Thus, a portion of subparagraph (c) has been
26 amended to recite:

27 when the Web page is loaded by the client device for rendering by the browser
28 program, causing the client device to execute the machine instructions that define how
29 to carry out the browser monitoring function,
30

1 Applicants have clarified that the browser monitoring function is defined by the machine
2 instructions. On page 7 of the Office Action, the Examiner has asserted that Bryant's code that is
3 submitted as part of the page is equivalent to applicants' machine instructions. Yet, notice that
4 applicants' machine instructions define how to carry out a browser monitoring program as recited in
5 step (c), in order to determine at least one performance metric. As explained above in connection
6 with applicants' traversal of the rejection of Claim 1, the function implemented by Bryant's code that
7 is submitted as part of the page does not actually determine at least one performance metric, but only
8 instructs *when* it should be determined.

9 For the reasons noted above, the rejection of Claim 21 as being anticipated by Bryant should
10 be withdrawn, since the claim is patentable over these references. Similarly, since dependent claims
11 inherently include what is recited in the independent claim on which they depend, Claims 22-24, 26,
12 29, and 31-34, are patentable over Bryant for at least the same reasons as set forth above in
13 connection with Claim 21.

14 Patentability of Independent Claim 35

15 Significant differences exist between the cited art and Claim 35 because the cited art does not
16 teach or suggest that the machine readable instructions that define a browser monitoring program
17 carry out a function used to determine at least one performance metric on the client computing
18 device. Independent Claim 35 is directed toward a memory medium and a portion of it recites (as
19 amended):

20
21 on which are stored machine readable instructions that define a browser monitoring
22 function, which when executed by a client computing device, cause the client
23 computing device to carry out the browser monitoring function.

24 Applicants have clarified that the machine readable instructions define the browser
25 monitoring program. Then, as also recited in the claim, the browser monitoring function, when
26 executed by the client computing device, cause it to determine at least one performance metric.

27 The Examiner has rejected this claim under the same rationale as Claim 21. As explained
28 above in connection with applicants' traversal of the rejection of Claim 21, the function implemented
29 by Bryant's code that is submitted as part of the page to the client does not actually determine at least
30 one performance metric, but only instructs as to *when* it should be determined.

1 For the reasons noted above, the rejection of Claim 35 as being anticipated by Bryant should
2 be withdrawn, since the claim is patentable over these references. Similarly, since dependent claims
3 inherently include what is recited in the independent claim on which they depend, Claims 36, 38, 40,
4 and 41, are patentable over Bryant for at least the same reasons as set forth above in connection with
5 applicants' traversal of the rejection of Claim 35.

6 Patentability of Independent Claim 42

7 Independent Claim 42 is directed toward a system for determining and collecting at least one
8 performance metric related to access of a Web page by a browser program. However, significant
9 differences exist between the cited art and the recited subject matter as the cited art does not teach or
10 suggest that the machine instructions that are included with the Web page perform a browser
11 monitoring function that determines said at least one performance metric. The Examiner has rejected
12 Claim 42 under the same rationale as Claim 14. However, as explained above in connection with
13 applicants' traversal of the rejection of Claim 14, Bryant's code that is included as part of the Web
14 page does not actually determine the performance metric of the fetch latency, but only instructs *when*
15 it should be determined.

16 For the reasons noted above, the rejection of Claim 42 as being anticipated by Bryant should
17 be withdrawn, since the claim is patentable over these references. Similarly, since dependent claims
18 inherently include what is recited in the independent claim on which they depend, Claims 43-45, 47,
19 50, 52, and 53, are patentable over Bryant for at least the same reasons as set forth above in
20 connection with Claim 42.

21 Dependent Claims Rejected under 35 U.S.C. § 103

22 The Examiner has rejected Claims 5, 8, 17, 25, 27, 28, 37, 46, 48, 49, and 54-56 as being
23 unpatentable over Bryant in view of U.S. Patent No. 5,732,218 (Bland et al., hereinafter referred to
24 as "Bland"). However, these claims ultimately depend from independent Claims 1, 14, 21, 35, and
25 42, respectively. Since dependent claims inherently include what is recited in the independent claim
26 on which they depend, Claims 5, 8, 17, 25, 27, 28, 37, 46, 48, 49, and 54-56 are patentable over
27 Bryant, in view of Bland for at least the same reasons as set forth above in connection with the
28 traversal of the rejection of Claims 1, 14, 21, 35 and 42.

29 The Examiner has rejected Claims 9, 20, 30, 39, and 51 under 35 U.S.C. § 103(a) as being
30 unpatentable over Bryant in view of Jia Wang (Survey of Web Caching Schemes for the Internet).

1 However, these claims ultimately depend from independent Claims 1, 14, 21, 35, and 42,
2 respectively. Since dependent claims inherently include what is recited in the independent claim on
3 which they depend, Claims 9, 20, 30, 39, and 51 are patentable over Bryant, in view of Jia Wang for
4 at least the same reasons as set forth above in connection with the traversal of the rejection of
5 Claims 1, 14, 21, 35, and 42.

6 In view of the preceding remarks, it should be evident that this application is in condition
7 for allowance and should be passed to issue without delay. Should any further questions remain,
8 the Examiner is invited to telephone applicants' attorney at the number listed below.

9
10 Respectfully submitted,

11
12
13 /sabrina k. macintyre/
14 Sabrina K. MacIntyre
15 Registration No. 56,912

16 SKM/RMA:elm
17
18
19
20
21
22
23
24
25
26
27
28
29
30